Static Keyword

1.Why do we need static keyword in Java explain with example.

The static keyword in Java is mainly used for memory management. The static keyword in Java is used to share the same variable or method of a given class. The users can apply static keywords with variables, methods, blocks, and nested classes. The static keyword belongs to the class than an instance of the class. The static keyword is used for a constant variable or a method that is the same for every instance of a class.
The static keyword is a non-access modifier in Java that is applicable for the following:

1. Blocks
2. Variables
3. Methods
4. Classes

// Example

```
class Test
{
        // static method
        static void m1()
        {
                System.out.println("from m1");
        }
        public static void main(String[] args)
        {
                // calling m1 without creating
                // any object of class Test
                m1();
        }
}
```

2.what is class loading and how does Java Program actually executes?

class loader is a part of the java runtime environment (JRE) that loads class into JVM .
i.e. class loader loads class into Memory dynamically
not one class loader loads all class , it depends on type of class , the class loader of perticular type is decided.
There are three types of class loader.
bootstrap classloader
extension classloader
system classloader
Methods used in class loader are -
1. loadclass(String name , boolean resolve)
2.final defineclass()

3.findClass(String name)
4.findLoadedClass(String name)
5.Class.forName(String name ,boolean initialize, ClassLoader loader)

3.Can we mark a local variable as static?

In Java, a static variable is a class variable (for whole class). So if we have static local variable (a variable with scope limited to function), it violates the purpose of static. Hence compiler does not allow static local variable.

4.Why is static block executed before the main method in Java?

A static block is a block of code with a static keyword. In general, these are used to initialize the static members. JVM executes static blocks before the main method at the time of class loading.

5.Why is a static method also called as a class method?

A static method in Java (also called class method) is a method that belongs to the class and not the instance. Therefore, you can invoke the method through the class instead of creating an instance first and calling the method on that instance.

6.What is the use of static blocks in Java?

static block is like a static field, but is a block that can contain an arbitrary number of lines of code. Just as static fields are initialized when the class is loaded into the JVM, static initializer blocks are executed just once, *when the class is loaded*. So they are guaranteed to be executed before any constructor is called, in other words before there are any instances of the class.

What is the use of them? There are many:

- They can be used to **initialize any static fields** which are too **complicated** to be set up with a one-line declaration. For example where there is a *process* of several steps to go through, where the value of a static field depends on something else. Another common case is setting up complex *structures* in a static field: for instance if you wanted to set up a map of country code to country name (eg “GB”: “Great Britain”) you could do this in a static initializer. Of course, data which could change would be better stored in a file than compiled into a class!
- They can be used to set up static **resources** in other ways, such as making sure other classes have been loaded (such as JDBC drivers or any other SPI implementation class), or reading in a properties file associated with the class. More usually this would be done lazily from a constructor, but if the checks to find out whether or not the resources have been loaded were heavyweight, you might not want to slow down the constructors with them. Using a static block guarantees that the code will only be run once.

- When initialization code could throw an **exception**, and you want to handle it. For example, you might want the class to load some resource into a static field, so that it's available to all instances. IOException is a checked exception, so you might want to catch it and rethrow it as a runtime exception, or just set a (static) flag to tell instances that the resource isn't available.
- The rules of Java say that if you're writing a constructor and want to call another constructor, this must be done first, before executing any other code. If you need to execute some code before this (most often this is because you want to *do something before the superclass's constructor* is called), you can't do it in the constructor. A static block is guaranteed to happen **first**. (But just to reiterate: only once, when the class is loaded, not every time an instance of the class is constructed.)

7.Difference between Static and Instance variable

Static variable	Instance variable
A static variable is a property of a class.	An instance variable is a property of an instance.
A static variable is created only once when the classloader loads the class.	An instance variable is created everytime an instance is created.
A static variable is used when you want to store a value that represents all the instances like count, sum, average etc.	An instance variable is used to store a value that represents property of single instance.

8.Difference between static and non-static members

Static variable

- A static variable is declared using the static keyword.
- Memory is allocated at the run-time when the class is loaded and only once.
  However, final keyword can be used to show that the value of a static variable remains the same after initialization.
- The static as well as non-static methods can access a static variable.
- A static variable is common to every object of the class.
- It is like a global variable, available to all.
- Above all, the class name is used to access the static variable outside the class.

Non-Static variable

- A non-static variable is declared as a regular variable.

- Memory is allocated when a new object is created. However, each time a new object is created new memory space is allotted which can lead to wastage of memory.
- The static methods cannot use a non-static variable.
- A non-static variable is accessible to its instance(where it is defined).
- It is a local variable and available to object of the class.
- Accessing a non-static variable is done using the reference to the instance where it is declared.